KIV/MKZ

Cvičení 7, 2016

L. Pešička

OBSAH

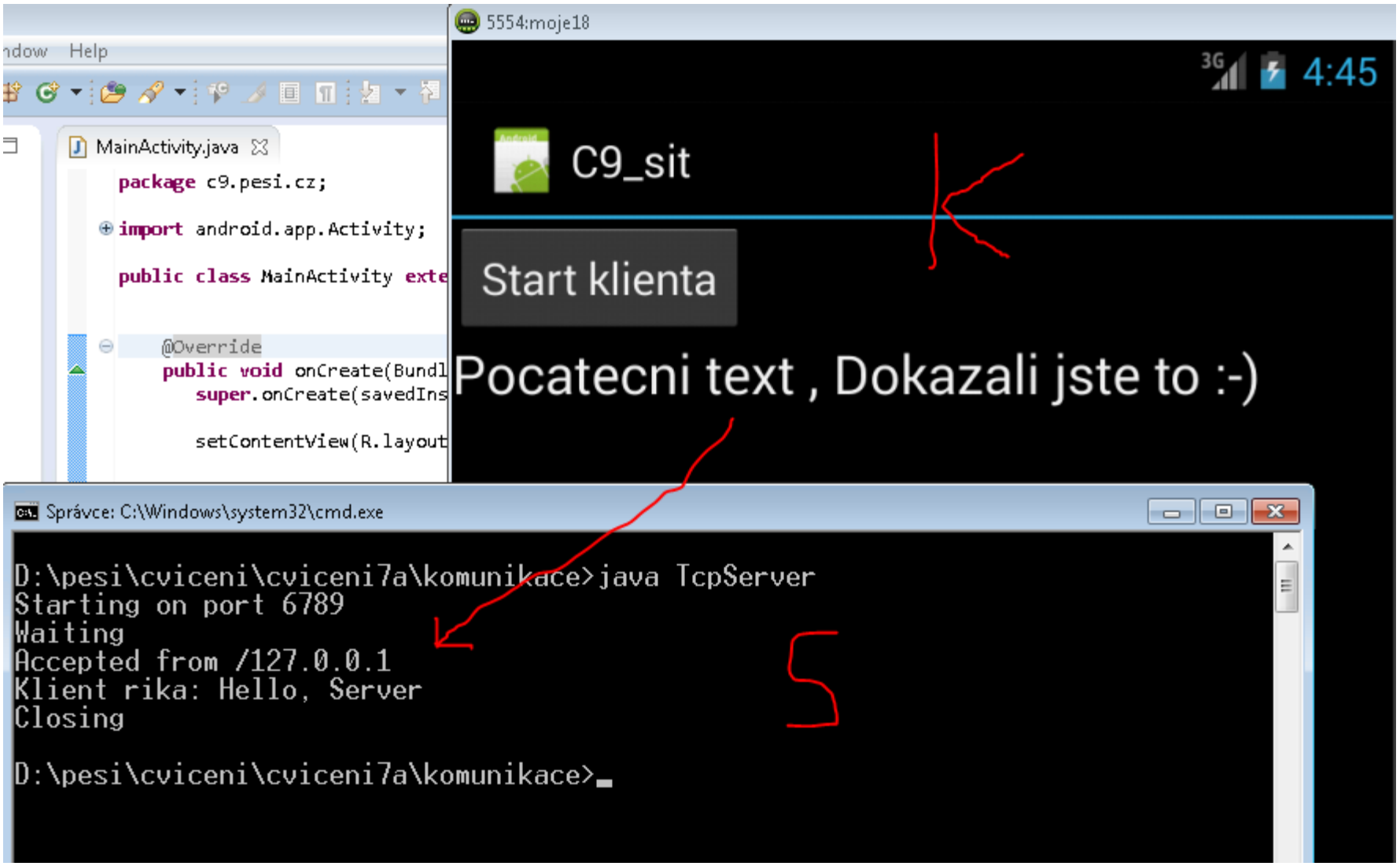
- síťování
- widget
  - Receiver
  - Receiver + služba

KLIENT / SERVER APLIKACE

- server v Javě na lokálním uzlu
  TCP, port 6789
- klient – aplikace v Androidu

- právo v manifestu pro přístup na Internet
  <uses-permission
  android:name="android.permission.INTERNET" />

- klient se připojuje na **10.0.2.2**, TCP 6789
  pro komunikaci s lokálním uzlem (stejným na kterém běží emulátor)


ndow Help
MainActivity.java
package c9.pesi.cz;
import android.app.Activity;
public class MainActivity exte
@Override
public void onCreate(Bundl
super.onCreate(savedIns
setContentView(R.layout
5554:moje18
3G
4:45
C9_sit
Start klienta
Pocatecni text , Dokazali jste to :-)
Správce: C:\Windows\system32\cmd.exe
D:\pesi\cviceni\cviceni7a\komunikace>java TcpServer
Starting on port 6789
Waiting
Accepted from /127.0.0.1
Klient rika: Hello, Server
Closing
D:\pesi\cviceni\cviceni7a\komunikace>

KLIENT

- síťování by nemělo probíhat v hlavním vlákně
- pro účely testování můžeme povolit (nastavit vhodnout politiku):

```
StrictMode.ThreadPolicy policy =
new StrictMode.ThreadPolicy.Builder()
.permitAll().build();

StrictMode.setThreadPolicy(policy);
```

KLIENT TCP - ANDROID

- v bloku try():

```
s = new Socket("10.0.2.2", 6789);
reader = new BufferedReader(new  InputStreamReader
(s.getInputStream()));
writer = new PrintWriter(new  OutputStreamWriter
(s.getOutputStream()));

String line;
writer.println("Hello, Server");
writer.flush();
line = reader.readLine();
tv.setText(tv.getText() + " , "+line);
reader.close();
writer.close();
```

SERVER TCP - PC

```
static Socket accept (int port) throws IOException {

    System.out.println ("Starting on port " + port);
    ServerSocket server = new ServerSocket (port);
    System.out.println ("Waiting");
    Socket client = server.accept ();
    System.out.println ("Accepted from " + client.getInetAddress ());
    server.close ();
    return client;
}
```

TCP SERVER - PC - II.

```
client = accept (DEFAULT_PORT);  // volá viz předchozí slide
PrintWriter writer;
BufferedReader reader;
reader = new BufferedReader(new
InputStreamReader(client.getInputStream()));
writer = new PrintWriter(new
OutputStreamWriter(client.getOutputStream()));

String line = reader.readLine();
System.out.println("Klient rika: " + line);
writer.println ("Dokazali jste to :-)");
writer.flush();
reader.close(); writer.close();
```

WIDGET - MINIAPLIKACE

Widget

aplikace zapouzdřená v jiné aplikaci (HomeScreen), přijímá periodické updaty

RemoteView

- View, které vykresluje jiná aplikace
- podmnožina klasických View

ORIENTAČNĚ:

1. manifet
2. widget_info
3. widget_layout
4. myshape (/res/drawable)
5. MainActivity (není v manifestu)
6. MyWidgetProvider
7. UpdateWidgetService (když využijeme službu)

WIDGET (WIDGET.ZIP)

3 XML soubory:

- vzhled widgetu (res/layout/widget_layout.xml)
- vzhled pozadí widgetu (res/drawable-mdpi/myshape.xml)
- chování widgetu (res/xml/widget_info.xml)

minimální interval aktualizace 30 minut
vliv na baterii zařízení
případně využít Alarm

uživatel může umístit více instancí widgetu
dle tutoriálu de.vogella.android.widget.example

WIDGET - WIDGET_INFO.XML

```
<?xml version="1.0" encoding="UTF-8"?>
<appwidget-provider
  android:widgetCategory="keyguard|home_screen"
  android:updatePeriodMillis="300000"
  android:minWidth="300dp" android:minHeight="72dp"
  android:initialLayout="@layout/widget_layout"

xmlns:android="http://schemas.android.com/apk/res
/android">
</appwidget-provider>
```

POZNÁMKA

- Widget může obsloužit funkcionalitu sám
- Často je ale vhodné, že pustí službu, která vykoná činnost a poté se sama zastaví

- Nejprve widget bez využití služby
- Později widget s využitím služby

WIDGET - ONUPDATE()

```
public class MyWidgetProvider extends AppWidgetProvider {

public void onUpdate(Context context,  AppWidgetManager  appWidgetManager,
      int[] appWidgetIds) {
ComponentName thisWidget = new ComponentName(context,
        MyWidgetProvider.class);
    int[] allWidgetIds = appWidgetManager.getAppWidgetIds(thisWidget);

    // Vytvoříme intent pro volání služby
    Intent intent = new Intent(context.getApplicationContext(),
        UpdateWidgetService.class);
    intent.putExtra(AppWidgetManager.EXTRA_APPWIDGET_IDS, allWidgetIds);

    // Update widgetu pomocí služby
    context.startService(intent);
  }
}
```

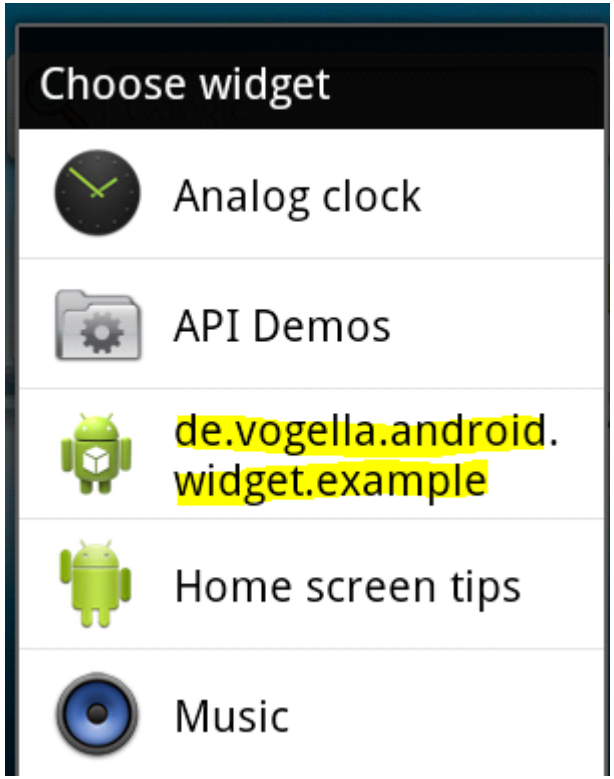
WIDGET - VYUŽÍVÁ SLUŽBU

widget může v metodě onUpdate volat službu, která provede update widgetu a může určit, kdy znovu aktualizovat

```
public class UpdateWidgetService extends
Service {

public void onStart(Intent intent, int startId) {
…
}
}
```

SIMULÁTOR - VÝBĚR A BĚH WIDGETU



můžeme umístit více instancí widgetu na plochu, kliknutím na libovolný z nich se změní náhodné číslo v obou instancích widgetu